AD-753 672

DISPLAYING COMPUTER GENERATED HALF-TONE PICTURES IN REAL TIME

Alan C. Erdahl

Utah University

Prepared for:

Rome Air Development Center
Advanced Research Projects Agency

August 1972

DISCLAIMER NOTICE

THIS DOCUMENT IS THE BEST
QUALITY AVAILABLE.

COPY FURNISHED CONTAINED
A SIGNIFICANT NUMBER OF
PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

RADC-TR-69-250
Final Technical Report
August 1972

DISPLAYING COMPUTER GENERATED
HALF-TONE PICTURES IN REAL TIME

University of Utah

Sponsored by
Defense Advanced Research Projects Agency
ARPA Order No. 829

Approved for public release;
distribution unlimited.

The views and conclusions contained in this document are those
of the authors and should not be interpreted as necessarily
representing the official policies, either expressed or implied,
of the Defense Advanced Research Projects Agency or the U. S.
Government.

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Computer Science University of Utah Salt Lake City, Utah 84112	Unclassified
	2b. GROUP

3. REPORT TITLE

DISPLAYING COMPUTER GENERATED HALF-TONE PICTURES IN REAL TIME

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

Alan C. Erdahl

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1972	60	5

8a. CONTRACT OR GRANT NO. AF30(602)-4277	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. ARPA Order No. 829	TR-4-14
c. Program Code Number 6D30	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-69-250
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES Monitore by Rome Air Development Center (EMIIO) Griffiss Air Force Base, New York 13440	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Wash DC 20301

13. ABSTRACT

This paper describes a system which will allow computer generated half-tone pictures to be displayed in real time. Half-tone or shaded pictures are valuable since they allow object surfaces to be displayed.

The picture must first be described in terms of its visible surface boundaries. This is a convenient form since the information required is produced by the hidden line algorithms used to generate the picture. Such a form usually represents a substantial savings in the number of bits required to describe a picture.

A display generator is described which will convert these edge descriptions into intensity values which can then be displayed on a raster scan device. A 512 by 512 point picture can be generated in less than 1/30 of a second. This picture can then be displayed directly on an oscilloscope or placed on a storage disk which will refresh a standard television set.

Ta

DD FORM **1473**
1 NOV 65

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Halftone						
Computer Graphics						
Display						
Raster						
Picture Compression						

WIDE INTEREST REPORTS

Selection and Printing

DATE _____

TITLE:	_____				

	WIDE INTEREST YES	NO	HOW MANY COPIES WILL WE SELL?	PRINT AT MICRO	CONTRACT	NUMBER OF COPIES TO BE PRINTED
Alexander						
Biorke						
Coyne						
Kudravetz						

COMMENTS: ☒ Bad Copy ☐ Unable to reach audience ☐ Other

NO! *Broken and lite print*

DET 1/16/73

INPUT INSTRUCTIONS

1. ACCESSION NUMBER _____

2. ☐ REGULAR PROCESSING: Disregard instructions below.

3. ☐ WIDE INTEREST: Follow instructions below.

4. FAS

5. Price at $ _____

PRINTING INSTRUCTIONS

1. PRINT 1-up

2. ATTENTION SPECIAL REPRODUCTION SUB UNIT: PRINT _____ COPIES.

3. PRINT AT ☐ Microreprographics, ☐ Contract 265-M or ☐ Other.

NTIS-181 (4/72)

DISPLAYING COMPUTER GENERATED
HALF-TONE PICTURES IN REAL TIME

Alan C. Erdahl

I

FOREWARD

This interim report describes research accomplished by Computer Science of the University of Utah, Salt Lake City, Utah, for the Advanced Research Projects Agency, administered by Rome Air Development Center, Griffiss Air Force Base, New York under Contract AF30(602)-4277. Secondary report number is TR 4-14. Mr. David A. Luther (EMIIO) is the RADC Project Engineer.

This technical report has been reviewed and is approved.

Approved: DAVID A. LUTHER
            Project Engineer

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

ABSTRACT

This paper describes a system which will allow computer generated half-tone pictures to be displayed in real time. Half-tone or shaded pictures are valuable since they allow object surfaces to be displayed.

The picture must first be described in terms of its visible surface boundaries. This is a convenient form since the information required is produced by the hidden line algorithms used to generate the picture. Such a form usually represents a substantial savings in the number of bits required to describe a picture.

A display generator is described which will convert these edge descriptions into intensity values which can then be displayed on a raster scan device. A 512 by 512 point picture can be generated in less than 1/30 of a second. This picture can then be displayed directly on an oscilloscope or placed on a storage disk which will refresh a standard television set.

INTRODUCTION

One of the fundamental problems with the electronic computer has been the interface between the man and the machine. Man must somehow specify what he wants the machine to do and the machine must be able to relate to the man what it has done. The knowledge about how to construct an interface between an intelligent but relatively slow man and a fast but unimaginative machine has in the past been tenuous at best.

The development of graphical displays is one attempt to provide a better interface between the man and the machine. Calliographic (line drawing) displays have been developed which are reasonably adequate for displaying line drawings and text. For some types of data a line drawing is not sufficient. Contour maps, architectural drawings and representations of complex objects often reveal more information when viewed as a shaded representation of solid surfaces. A shaded surface can be displayed by a calligraphic display in several ways:

1. by filling in objects with a number of parallel lines
2. by constructing surfaces from a number of individual dots.

Both of these methods require relatively long display lists. The time required to produce one frame of reasonable complexity is usually such that the output can only be viewed by a time exposed photograph. Such a system, although

1

possibly useful, is hardly the ideal system for interaction between the man and the computer.

The design of a half-tone display system is presented in this paper. Such a system will produce shaded representations (half-tones) of desired pictures. The input to the system may be digitized intensities from a natural picture, a mathematical description of objects in 3 space, or combinations of the two. The output may be viewed directly on a raster scan device or photographed on an oscilloscope.

For the purpose of this paper the half tone display system will be divided into three parts:

1. The computer (or other device) which will define the picture.

2. The hardware (digital logic) which is designed to aid the computer in generating the picture will be referred to as the Display Generator. The purpose of the Display Generator is to take a coded description of the picture from the computer and expand it into digital intensity values for each part of the picture.

3. The devices which display and/or record the picture. The storage disk used to refresh the display and the digital to analog converters will be considered as part of the displays.

This paper will deal primarily with the Display Generator.

Section 1

THE COMPUTER

The role of the computer may vary widely depending upon the type of picture to be displayed. The description of the picture may be a picture itself. In this case the picture would be digitized by a scanning device, possibly processed in some way and then sent to the Display Generator. The case which will be considered in more detail is when at least part of the picture is described in a mathematical sense. In this case a great deal of processing may be required before the picture can be displayed.

The processing which may be required on objects before they can be displayed will involve the determination of which objects or parts of objects are visible and then formating the information about these visible surfaces. Certainly the computer may be used to define and manipulate the objects in the first place but these functions will not be considered as part of the display system. The determination of which parts of an object or objects are visible and which are hidden is often referred to as the "hidden line problem." This problem has been solved by a number of people with varying degrees of success. It is interesting to note that in at least two of the fastest algorithms known to the author (2,3) geometrical edge boundaries are used in describing and processing the surfaces. This is one reason edge boundaries will be used throughout the Dis-

play Generator to describe visible surfaces. One of the hidden line algorithms will be considered briefly.

The Warnock Hidden Line Algorithm (3)

The method Warnock uses to solve the hidden line problem is basically very simple. His algorithm considers the objects to be projected onto a viewing screen. It then asks the question, "Is this picture 'simple enough' to be displayed?" "Simple enough" is defined to mean that there is at most one edge or intensity discontinuity in the picture. If the answer is 'yes', then either there is nothing in the picture or else simple calculations can be used to find the intensities of all points in the picture. If the answer is 'no', the picture is divided into four parts and the process is continued on each part. This process of subdivision continues until the sub areas become "simple" or reach the resolution limit of the picture. The beauty of this algorithm is that the time required depends primarily upon the complexity of the visible surfaces rather than on the number of surfaces defined in the picture.

For Warnock's algorithm, complex surfaces are first approximated by flat triangles. The result of the algorthm is the geometrical description of the visible boundaries between flat surfaces. This description and the associated intensity information are sufficient to describe all the visible objects in the picture.

Once the computer program has defined the edge boundaries it could interpolate between the boundaries and thus

calculate the intensity at every point in the picture. In
order to reduce the amount of computation and the band-
width required of the computer a special hardware device
has been designed to do this interpolation. This device
is called the Display Generator. The operation of the
Display Generator will be described in a later section of
this paper.

Boundary Definitions

A boundary definition consists of the description
of the geometric boundary between two visible surfaces and
the intensity function on one side of that boundary. Since
all surfaces are approximated by flat triangles, the bound-
ary definitions have special properties.

1. All geometric boundaries must be straight lines.
2. The intensity function connected with a boundary
    definition relates to the area to the right of
    that boundary.
3. No two boundaries may cross. When two boundaries
    meet, at least one must end. This requirement
    is necessary in order to keep the display list
    in order. This is a requirement of the boundary
    definitions produced by the computer, not of the
    objects themselves.

For natural pictures the left edge of the picture becomes
the geometric boundary and the intensity function is
specified by listing the intensity values at every point

in the picture.

The boundary definitions will be one of three types depending upon the type of intensity function with which they are associated.

1. Constant intensity - areas which have a constant intensity value. Blank areas of the picture could be black or have some pre-assigned intensity level.

2. Linear intensity - areas where the intensity can be approximated by a linear (first order) function of X (horizontal) and Y (vertical) position. Linear functions can be of two types:

    (a) 2-D When the intensity is a known linear function of both X and Y. That is, the intensity function is known in two dimensions thus 2-D.

    (b) 1-D When the intensity is a known linear function only in the X direction. The intensity in the Y direction may be something other than linear or maybe unknown.

3. 0-D Where the intensity function is non-linear in the X direction or where the intensity function is unknown. In this case the intensity value must be specified for each point. thus 0-D. Natural pictures are of this type.

The boundary definitions thus have the purpose of defining the discontinuties of the intensity function.

6

In order to completely describe a plane surface its boundaries and the intensity values of its surface must be defined. If it is assumed that the entire picture is covered by surfaces, then every point on the picture is defined as being on some visible surface. This is equivalent to assuming that there is a large screen behind the object being viewed. This background screen may of course be black. Thus the picture may be completely defined by including only the boundaries where an imaginary beam scanning the picture would enter each surface. Every time it exists from one surface, it enters another and thus that boundary is defined by implication.

The information required for each 2-D boundary definition is:

a. Geometrical information

Xo   The X value of the first place the beam will encounter the boundary

Yo   The Y value of the first place the beam will encounter the boundary

$\Delta X/\Delta Y$ The inverse of the slope of the boundary. This is calculated from the beginning (Xo,Yo) and end (Xe,Ye) points of the boundary. The beginning and end points must lie on scan lines but Xo and Xe can have fractional values between the discrete points on the scan line. $\Delta X = Xe - Xo$ and $\Delta Y = Ye - Yo$.

Yc The total number of scan lines for which the
boundary is defined. (yc = Ye-Yo)

b. Intensity information

Io The intensity at Xo,Yo

dI/dX The variation of intensity with respect to X
along the scan line.

dI/dY The variation of intensity with respect to Y
along the boundary line.

The information required for each 1-D boundary definition
is:

Xo - The X value on the scan line where the boundary
starts

Yo - The scan line for which the boundary is defined

Io - The intensity at Xo,Yo

dI/dX - The variation of intensity with respect to X

Either the 2-D or 1-D definitions can represent a constant
intensity area by having dI/dX and dI/dY equal to zero.

Areas of 0-D intensities could be represented as
degenerate cases of 2-D or 1-D boundaries but the amount
of useless information required would be excessive. Thus
for the 0-D case only one boundary is defined (this may in-
clude the entire picture) in the above manner. For all
points within this boundary only the actual intensity
values are specified. These values are packed into words
to make use of all the bits available.

INFORMATION FORMAT

Input Format *

Pr - priority bit
DP - Disable Pre-Add
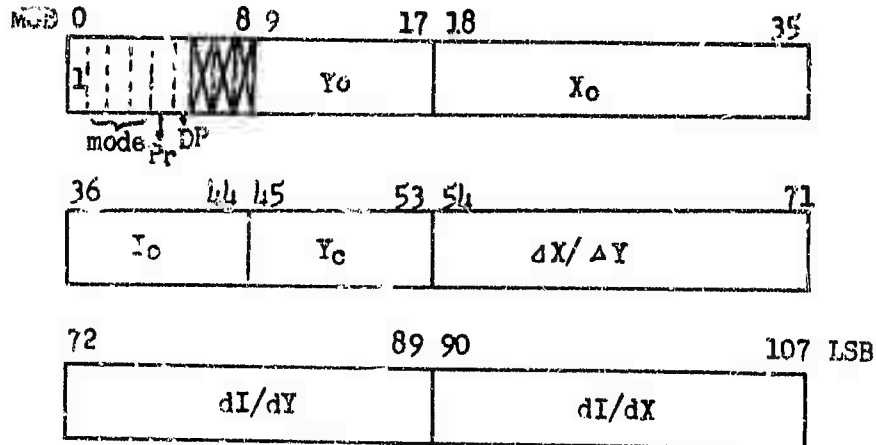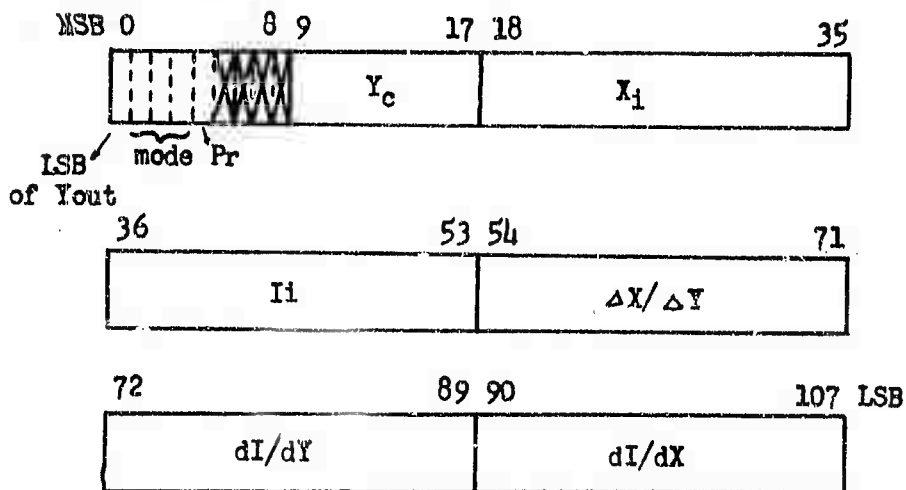
* This is the instruction format used by the computer to specify boundary definitions to the Display Generator.

Memory Format **

** This is the instruction format used in the update- Memory A loop. (Information which is retained for more than one scan line).
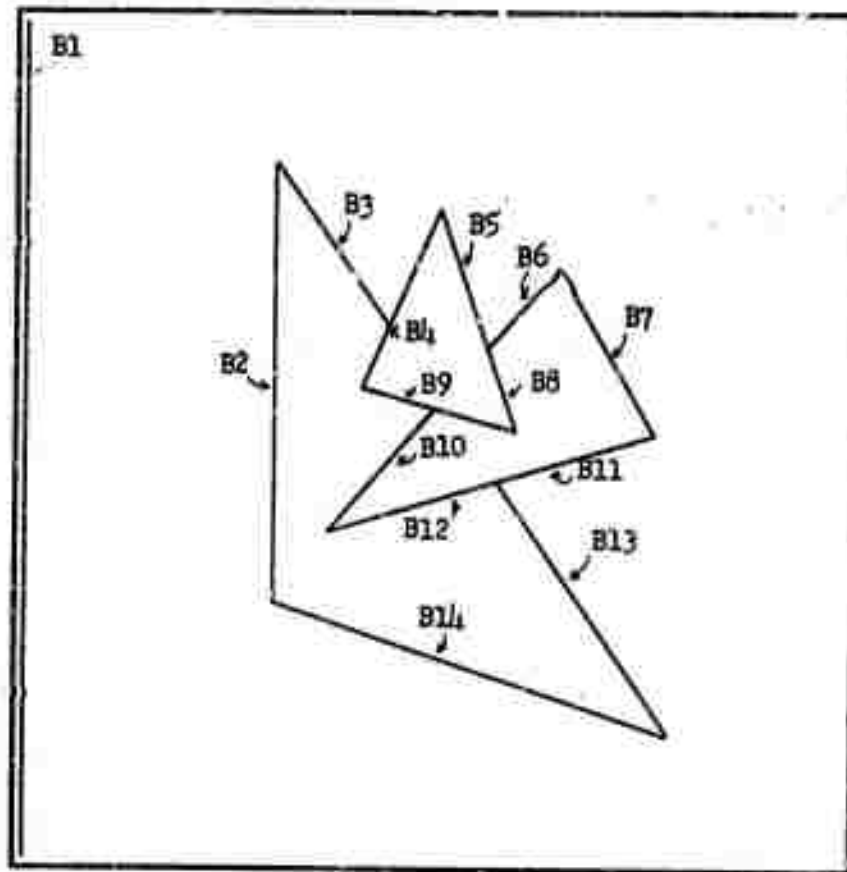
Figure 1

The ordering of the boundary definitions is very im-
portant. In order to know when the scan leaves one area
the X,Y coordinates of the next one to be entered must be
known. If the boundaries were not initially ordered, a
search of all the boundary definitions for the entire
picture would be required to determine which boundary will
be crossed next. This search would require a great deal
of storage and a lot of time and/or hardware. If however,
the boundary definitions are stored according to the order
that the raster scan will cross them, only the next bound-
ary definition in the list will need to be referenced to
determine where the present area ends and the next begins.
In order to preserve this ordering, boundaries must be pro-
hibited from crossing. Fig. 2 shows typical boundaries.

Precision Requirements

The number of bits required for each boundary definition
depends upon how accurately the picture is to be described.
If too much accuracy is required in the description, the
number of bits required for each picture and the bandwidth
required to display that picture becomes excessive. If not
enough accuracy is required, the final picture will lack
detail and information may be lost. Thus the first decision
which must be made is the required resolution of the picture.

Since the final picture will be displayed by a raster
scan device, there will be an integral number of scan lines.
In this system there will also be an integral number of

10

Edge Boundaries

Raster Scan Direction

This picture shows the edge boundaries required to
define a simple picture. Each boundary definition
contains information about the intensity of the
area to the right of the boundary.

Figure 2

points along each scan line. Each point will have an individual intensity value. Thus the entire picture will be composed of a large number of individual points of varying intensity.

The display device described in this paper will assume that each picture is composed of 512 scan lines, each containing 512 points. This resolution was chosen since such a picture has reasonably good quality, yet the bandwidth required is not excessive. A standard television has from 485 to 525 horizontal lines, each containing 400 to 600 elements. Thus a 512 x 512 picture has about the same number of points as a standard television.

There are many advantages of having a standard byte size with all parameters consisting of a small number of bytes (1 or 2). This will facilitate the formating job required of the computer. Because of the resolution of the picture, the byte length is chosen to be 9 bits.

Since there are 512 scan lines, Yo and Yc which refer to scan lines require 9 bits each. $\Delta X/\Delta Y$ defines a boundary slope which can vary from vertical to horizontal in increments of ± 1 part in 512. In order to remain consistent with the 9 bit byte, $\Delta X/\Delta Y$ will be specified by 18 bits, 9 bits for the fractional part and 9 bits (including the sign) for the integer portion. (This only restricts horizontal lines which completely cross the screen.) Xo also requires 18 bits to avoid rounding errors in the updating calculations.

The number of bits required to define the intensity
value at each point is not easy to determine. Although the
eye may not be able to distinguish more than 8 shades of
gray on a given display, it can often detect boundaries
between 64 or more shades of grey on the same display.
In order to make the intensity step size small enough so
that surfaces will appear to be smooth and to remain
consistent with the geometric definitions 9 bits will be
used to specify the intensity at each point. Thus Io
can be specified as 9 bits in the boundary definition but
must be 18 bits when it is updated from one scan line to
the next. dI/dX and dI/dY will require 18 bits. Nine bits
also need to be reserved for instructions and specification
of the boundary type. Thus each boundary definition will
require 108 bits of 12 bytes of information.

Lines

In any display system it is extremely convenient to
have the capability of drawing lines. It would seem that
if a device could display solid objects, lines should be
easy since a line is a degenerate form of a surface. Un-
fortunately lines cannot be produced the same way surfaces
are. The display is essentially a large grid produced
by the raster scan with each square on the grid having
a distinct intensity. A line is produced by intensifying
a number of these squares. Horizontal and vertical lines
are easy to draw but lines at other angles must be con-

structed from short horizontal and vertical segments.

Solid objects in the display system are defined in terms of their boundaries. Since a line is a boundary, it seems probable that all lines could be drawn simply by intensifying the first square that the boundary crosses on each scan line. This method works reasonably well if the line has a slope of 45° or more. If the line has a slope of less than 45° several points must be intensified on each scan line to produce the effect of a continuous line. (see figure 3) The number of points which should be intensified may not be the same on each scan line. A second consideration which should be kept in mind is that positive sloped lines must not be biased with respect to negatively sloped lines. If this is allowed to occur, lines fail to meet and may overlap.

Probably the simplest and most straightforward method of drawing lines is to intensity every square through which the line passes. (see figures 4 and 5) This scheme is relatively easy to implement and produces lines which are unbiased with respect to the rest of the picture. At each scan line the horizontal value of the line is known (Xi). The horizontal value of the line at the next scan line (Xi+1) can be calculated by adding $\Delta X/\Delta Y$ to Xi. For lines with positive slopes, all squares from Xi through Xi+1 are to be intensified. For lines with negative slopes ($\Delta X/\Delta Y$ is negative) all squares from Xi through Xi-1 are

Illustrations of Lines

Lines produced by intensifying the first point where the raster scan crosses the line boundary.

Figure 3-A

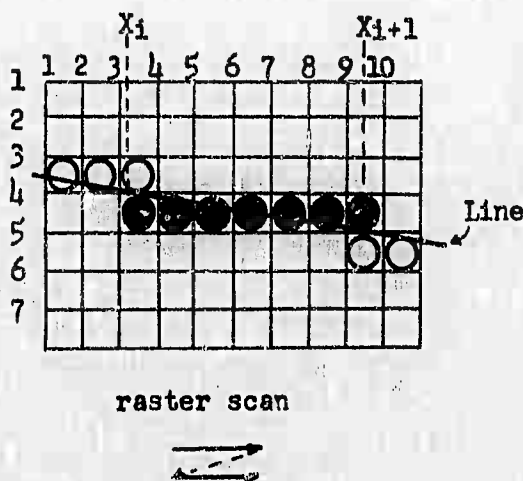In the angle θ is greater than 45°, this method produces a fairly good line.

Figure 3-B

If the angle θ is less than 45°, lines produced by this method are not very good. The smaller the angle θ becomes, the worse the line appears.

Lines produced by intensifying every

square through which the boundary passes.

Positive Sloped Lines

raster scan

$X_1 = 3.2$

$X_{1+1} = 9.5$

$\Delta X/\Delta Y = 6.3$
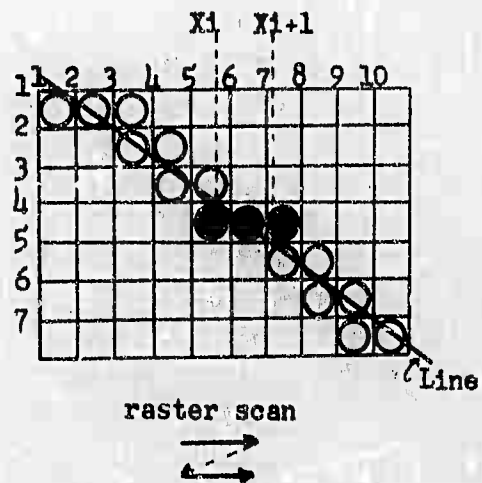
On scan line four:

First square intensified $= 3$

# of squares intensified

$int^{(X_1 fr + abs(\Delta X/\Delta Y)+1)}$

$(.2 + 6.3 + 1)$

$= 7.5$ thus 7 squares

are to be intensified

$X_1 = 5.7$

$X_{1+1} = 7.2$

$\Delta X/\Delta Y = 1.5$

On scan line four:

First square intensified $= 5$

# of squares intensified

$int^{(X_{1fr} + abs(\Delta X/\Delta Y)+1)}$

$(.7 + 1.5 + 1)$

$= 3.2$ thus 3 squares

are to be intensified

Figure 4

16

Lines produced by intensifying every

square through which the boundary passes

Negative sloped line
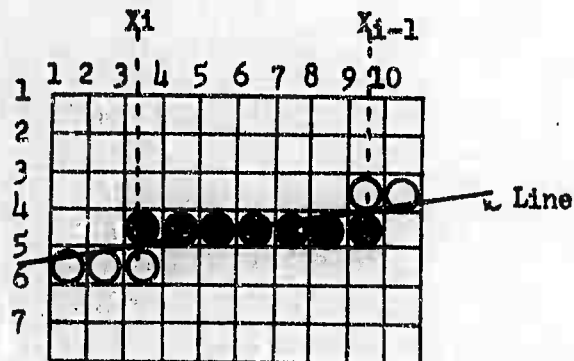
Raster scan direction
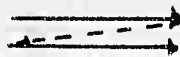
$Xi = 3.2$

$Xi-1 = 9.5$

$\Delta X/\Delta Y = -6.3$

On scan line four:

First square to be intensified $= Xi_t = \underline{3}$

Number of squares to be intensified

$$\lfloor (Xi_f + |\Delta X/\Delta Y| + 1)$$

$$(.2 + 6.3 + 1)_t = \lfloor 7.5 = 7$$

Thus 7 points are to be intensified
on scan line 4

Figure 5

to be intensified. This difference between the procedures
for positive and negative lines results from the fact that
negative lines must be updated before they are displayed.
In either case the first squares to be intensified is
Xi+(truncated value). The number of squares to be inten-
sified is calculated by adding the fractional part of Xi to
the absolute value of $\Delta X/\Delta Y$, then adding 1 and truncating
the result. The number of squares to be intensified =
Int(fract of Xi+ $|\Delta X/\Delta Y|$ + 1).

Section 2

DISPLAY GENERATOR

The purpose of the Display Generator is to expand the
coded boundary definitions into point intensity values.
This expansion is accomplished by interpolating between
the boundary definitions. Since the output may be
displayed on a television set, the Display Generator
msut be able to produce the point intensities at television
rates. *This* means that each scan line (512) points must
be produced in 53.3 u sec. or a new intensity value every
104 nsec (10^{-9} sec.)[1] In this paper a cycle will be
considered to be 100 nsec.

The Display Generator is designed to operate in several
ways to accommodate various types of display devices. The
Display Generator may be clocked internally or externally.
An external device may request intensity points singly at
any rate up to one every 100 nsec. The Display Generator
may also be allowed to run at its own rate and be syn-
chronized on each scan line or each frame by the external
device. If for any reason the Display Generator cannot
keep up with the display device or the memory capacity
is exceeded, it can stop and be restarted at a later time
by the display device. It can also be allowed to continue
and catch up as soon as possible. If it is allowed to
continue, part of the picture may be lost.

Operation of the Display Generator

Information is transferred from the computer or input device through an interface to an input buffer memory. In this memory the information is stored as 108 bit words. Each of these words is either a boundary definition or a number of individual intensity values. The boundary definitions are transferred to Register B. During this transfer some processing is done on boundaries having a negative $\Delta X/\Delta Y$. The words containing intensity information (O-D data) are transferred to Register D1 or Register D2 as they become empty. The words are split into 9-bit intensity values and used when required by an O-D boundary definition.

The boundary definition in Register B contains new information while the boundary definition in Register A contains updated information from the previous scan line. These two paths of information must be merged in the proper sequence to maintain the ordering of the boundary definitions. A decision is made as to which one occurs in the picture first and that boundary is transferred to Register C as soon as Register C becomes empty. This decision process is called a merge operation. If Register A is chosen by the merge, it is refilled from Memory A. If Register B is chosen, it is refilled from the Memory Buffer of the Input Buffer Memory.

The boundary definition in Register C represents the next boundary which the raster scan will cross. As soon as the X, Y position of the beam (as determined by the X and Y counters) is equal to or greater than the X, Y value of the boundary, the contents of Register C is transferred to the Instruction Register. The boundary definition in the Instruction Register controls the intensity function of the beam. This present intensity function remains in effect until another boundary is crosses and a new boundary definition is placed in the Instruction Register.

The X, Y counters which control the beam position continually move the beam to form a raster scan across the screen of the display. These counters can be stopped at the end of a scan line or between frames in order to synchronize with an external storage device or display. The counters may also be stopped if the bandwidth of the input channel is unable to keep up with the display. These counters will continue when signaled to do so by the external device or when the required information has been transmitted from the input device.

After a boundary description is used in the Instruction Register, it is transferred to the Reject Register. In the Reject Register the decision is made as to whether or not the boundary definition is to be updated for use on the next scan line. If the Yc portion of the word is zero, the

word is discarded. If Yc is not zero, the boundary definition
is updated and sent to the Store Register. The Store
Register loads the boundary definition into Memory A where
it is stored until it is required in the next scan line.
Memory A is a FIFO (First In - First Out) list which keeps
Register A filled.

Computer Interface

The job of the computer interface is to accept data
from a computer or other input device, arrange it into 108
bit words, and transfer these words to the Input Buffer
Memory. This interface must be able to send and receive
all signals required for transmitting data from the input
device to the Display Generator.

The exact nature of the inputs to the interface de-
pends upon the type of device which sends the information.
In any case, once an entire word (108 bits) has been received,
the interface flag (flag 1) will be set. During the first
free cycle of the Input Buffer Memory, this word will be
transferred to the memory and the flag cleared. The clearing
of this flag is the signal to the interface that it can
start receiving the next word from the input device.

Input Buffer Memory

The Input Buffer Memory is an integrated circuit memory
addressed as a FIFO (First In - First Out) list. Its purpose

is to provide temporary storage for the display information. The bandwidth requirement of the Display Generator may vary drastically for different parts of the picture. At some points in the picture a new boundary definition may be required every cycle (100 nsec). During other parts of the picture, no new information may be required for several milliseconds. This memory allows the input requirements to be averaged over a period of time and thus reduce the required bandwidth of the input device.

This memory must be able to read a word from the memory or write a word into the memory in one cycle time. The length of each word is 108 bits. Words are read out of the Buffer Memory into the Memory Buffer Register. This read operation has priority over the write operation and the Memory Buffer Register will be kept full as long as there are any words contained in the Buffer Memory.

The size (number of words) of the memory will depend upon the nature of the picture to be produced and the bandwidth of the input device. Sixteen to 64 words might be considered as a reasonable size for this memory.

Registers B, D1, D2, and the Pre-Add Operation

Information is passed from the Memory Buffer Register to either Register B, D1, or D2. If the word is a boundary definition (as indicated by having the MSB = 1) it is transferred to Register B. If the word is a data word (MSB = 0)

it is transferred to one of the D registers. These transfers take place whenever the next register is empty. Transfers between several registers may occur at the same time. Because of the method used to display lines, all boundary definitions having a negative $\Delta X/\Delta Y$ must be updated before they are first displayed. This updating involves the addition of $\Delta X/\Delta Y$ to Xo. This addition is done during the transfer from the Memory Buffer Register to Register B. If the slope of the boundary is positive or if bit-5 of the boundary definition is 1 (disable pre-add) the addition circuits are bypassed.

Data words are transferred to Register D2 if it is empty. If Register D2 is not empty, the data word will be sent to Register D1 as soon as that register becomes empty. (Registers are defined to be full if the associated flag is a 1 and empty if it is a 0). Register D2 is a shift register which shifts its contents 9 bits at a time. The intensity values are shifted to the left and transferred out in 9-bit bytes as required. Since the number of intensity points associated with a given boundary definition may not fit evenly into 108 bit words, the counter controlling the shift register D2 is reset after each O-D boundary.

Merge Operation

The Merge operation is a sorting operation between two converging information paths. Register A contains information

from the previous scan line which has been updated for the present scan line. Register B contains new information which may or may not apply to the present scan line. The boundary definition contained in either Register A or Register B which will be crossed first by the raster scan is to be transferred into Register C. This transfer occurs as soon as Register C becomes empty.

Decision Procedure

Register A is a candidate for the Merge if it pertains to the present scan line (bit 0 of Register A = the LSB of the Y counter). Since all boundary definitions which appear in Register A must apply either to the present scan line or to the next scan line, only one bit is required to distinguish between the two. The boundary definition in Register B is a candidate if its Yo is less than or equal to the Y counter. (Using a less than or equal test rather than a strictly equal test allows boundary definitions which are out of order to be flushed through the system without destroying the entire picture). If both Registers A and B contain candidates for the Merge, the one with the smallest value of X is chosen. If Register B is empty the Merge operation cannot continue and the entire display may need to be stopped.

Register C

Register C contains the next boundary definition which the raster scan will cross. The scan position is controlled

by the X and Y counters. When the Xo value in Register C
is equal to or less than the X counter, the raster scan has
crossed that boundary. As soon as the scan crosses the
boundary, the boundary definition is transferred into the
Instruction Register where it controls the intensity of
the beam.

Instruction Register

The boundary definition held by the Instruction Register
controls the intensity function produced by the Display
Generator. Each boundary definition will remain in the
Instruction Register until the raster scan crosses another
boundary and a new boundary definition is transferred from
Register C. Even though a boundary definition may be
retained in the Instruction Register for some number of
cycles (a maximum of one scan line) it is also passed on to
the Reject Register one-half cycle after it arrives in the
Instruction Register.

The three mode bits and the priority bit determine the
nature of the intensity function. Depending upon the
specified mode, information may be transferred from the
Instruction Register to any of several registers and the
intensity output may be taken from either the Shader Register
D2, or the Constant Intensity Register. The function of
each mode is explained below.

Mode

000 - OFF - This mode is not decoded and thus the intensity produced is zero (black). This mode could be used to implement a desired intensity function at a later date.

001 - Line Mode - This mode draws a line by intensifying a number of points on each scan line. This mode is the same as mode 101 (constant intensity for count) except that the count is defined by the values of Xi and $\Delta X/\Delta Y$ as is described in a previous section. (p. 14)

$$\text{count} = 1 + \text{Int}(\, Xi_{fract} + |\Delta X/\Delta Y| \,)$$

010 - Linear Shading - Ii and dI/dX are initially sent to the shader. The intensity for each point is defined by the 9 MSB of Register Is. During each cycle dI/dX is added to Register Is. This sum becomes the new value of Is. This function produces a linear intensity ramp across the scan line. The slope of the ramp is dI/dX and the initial value of the ramp is Ii.

011 - Return to Mode 010 - This mode transfers control of the intensity function to the shader (mode 010) but does not reinitialize the registers in the Shader. The Shader continually adds dI/dX to Is regardless of what mode the Display Generator is in. Mode 011 thus returns control to an intensity function which was initiated earlier.

27

100 - Constant Intensity - This mode sets the output intensity
equal to the 9 MSB of Ii as specified by the boundary
definition.

101 - Constant Intensity for Count - In this mode the
intensity will be equal to the 9 MSB of Ii (same as
mode 100) for N points or until the next boundary is
crossed. N is defined by the 9 MSB of dI/dX. If
another boundary is not crossed after N points, control
will revert to mode 010 (linear shading). If the priority
bit is 1, the intensity will remain equal to the constant
value for N counts regardless of how many boundaries
are crossed.

110 - 0-D Mode - This mode is used to display point intensity
functions. For each point between this boundary and the
next one, a 9 bit intensity value is taken from Register
D2.

111 - End of Frame Sync. - This mode stops the display.
The display can be restarted by an external sync. signal
or by the system clear signal.

Priority Bit - The priority bit applies only to modes 011
and 101. If the priority bit in the boundary definition
(bit 4) is a 1, the count register has priority over boundaries.
This means that the specified intensity will be generated
for the specified number of points regardless of how many

28

boundaries are crossed in the process. If the priority bit
= 0, boundaries have priority over the count. In this case
the specified intensity lasts for either the specified
number of counts or the next boundary crossing, whichever
comes first. The priority bit does not alter the merge or
update operations. It may be used for drawing characters
or lines over other boundaries.

Overwrite Function

The linear shading function is performed by the Shader
and is independent from other functions. Once the shader
is initialized it will continue to function (adding $\Delta X/\Delta Y$
to Is each cycle) as long as the display is running. This
independence allows a constant, O-D or line function to
"overwrite" a linear intensity function. Modes 101 and
001 automatically revert to mode 010. Modes 100 and 110
can be terminated and control returned to mode 010 by a
mode 011 boundary. This "overwrite" feature makes it easy
to paint lines, symbols or natural pictures over other areas.

The overwrite feature and the priority bit allow the
use of two levels of priority. The natural question which
arises is: "Why not allow three or four levels of
priority?" Admittedly, such multiple levels of priority
could be useful. In general, if N levels of priority (or
anything) are provided, it is easy to show that N + 1

levels would be useful for some purpose. The additional logic circuits which would be required appear to make such extensions impractical.

Reject Register and the Update Processor

Following the Instruction Register, the boundary definitions are transferred to the Reject Register. If the Y_c portion of the word is equal to zero, the boundary definition will be discarded. Y_c refers to the number of scan lines from which a boundary definition is to be used. Each time the boundary definition is updated, the value in the Y_c part is decreased by one. Thus when Y_c equals zero, the boundary is terminated. If $Y_c \neq 0$, the boundary definition will be updated for the next scan line. This updating process involves four operations.

1. Subtract 1 from Y_c.
2. Set bit 0 to the LSB of Ycount (Y out).
3. Add $\Delta X/\Delta Y$ to X_i - this sum becomes the new X_i.
4. Add dI/dX to I_i - this sum becomes the new I_i.

The effect of this updating process is to interpolate the boundary definition in the Y direction.

Store Register, Memory A and Register A

The Store Register contains the updated boundary definitions. These definitions are stored in Memory A until they are needed for the next display line. Information from Memory A is transferred into Register A where it is

merged with the incoming boundaries and displayed.

Memory A is a fast integrated circuit memory. It must be able to read one word (108 bits) from one location and write another word into another location—all in one cycle (100 nsec). The maximum number of boundary definitions which could be defined for one scan line is 512. This case could occur only if there was a separate boundary definition for every point on a scan line. For most types of pictures, 256 storage locations would be sufficient.

Memory A, like the Input Buffer Memory, is addressed as a FIFO list. Three counters—a read counter, a write counter, and a number counter—are used to implement this addressing. The read counter is incremented every time a word is read from the memory. The write counter is incremented every time a word is written into the memory. The number counter is incremented when a word is written into the memory and decremented when a word is read from the memory. A new word may not be written into the memory if the number in the number counter equals the size of the memory. Likewise a word may not be read from the memory if the number counter equals zero. If the number of boundary definitions on any scan line exceed the size of the memory, some picture information may be lost.

Section 3

DISPLAY DEVICES

The X, Y and Z (intensity) values for the picture
are calculated by the Display Generator and then sent to
the Display Device. There are many things which could be
done with the completed picture. In this paper only
two display methods will be considered.

1. Making a hard copy of the picture

2. Viewing the picture directly

Photographic Copy

A photograph can easily be made with most good oscillo-
scopes, a camera and a mount. The X, Y and Z outputs of
the Display Generator are connected to digital to analog
converters and the resulting voltages move the beam and
control its intensity. The picture need only be displayed
once to expose the film. If the bandwidth of the input
device is not sufficient, the display can be stopped and
restarted at any time. The only requirement is that the
intensity is blanked during the time the display is stopped.
Color photographs can be made on a regular oscilloscope
with the use of colored filters and multiple exposures.
Motion pictures can be made synchronizing the shutter and
advance mechanism with the display. For high quality
photographs it is essential that the X, Y and Z charact-
eristics of the oscilloscope inputs and D/A converters are
linear. It is also important that the beam spot size be

32

kept fairly constant over the intended range of intensities.

The picture could also be produced on any of a number of devices which mark, burn or otherwise affect a surface in a raster scan fashion.

Direct Viewing

Direct viewing of the picture in real-time may be more difficult than merely taking a photograph of the picture. It might be possible to view the picture on a storage display oscilloscope if such a scope could be constructed so as to allow enough control of the intensity. It is desirable, however, to be able to produce a standard television picture since terminals for displaying such pictures are inexpensive and easily obtainable. One good way to achieve this goal is to have auxiliary storage between the Display Generator and the television. A storage disk or similar device is very well suited for such use. Using such a device each picture need be produced only once and stored on the disk. One or more television sets can then be refreshed directly from the disk. Once the picture is generated and stored it can be viewed as long as desired without requiring any further action from either the input device or the Display Generator.

Since the disk storage can hold several frames, one frame can be displayed while the other is being generated. Even though the picture must be refreshed from the disk at a rate of 30 frames/sec., the generation of the picture can

33

take as long as required. Although the disk moves at a constant rate, with proper addressing individual points can be read from or written on the disk.

Section 7

ALTERNATIVES

The Display Generator described in this paper has not yet been built. The economic practicality of building such a device has, in the past, been doubtful. The present trend of higher logic speeds and lower cost per logic element, should make a device such as this one practical in the near future if it isn't already.

One of the major expenses of the Display Generator is the large integrated circuit memory. Constructing such a memory from indivdual flip-flops is too expensive to be practical. Recent advances in the state of the art of integrated circuits promise to make such memories not only readily available but also relatively inexpensive.

A second consideration in the evaluation of the Display Generator is the anticipated efficiency of such a device. This must be considered in terms of the amount of computer time saved and the reduction in the required I/O bandwidth of the computer. There are 512^2 or 262,144 points in each picture. If the intensity value for each point were specified, it would require 2,359,296 bits/picture (9 bits/point). This same number of bits could be used to specify 21,845 boundary definitions (108 bits/boundary definition). Thus the bandwidth reduction of the Display Generator depends upon the complexity of the picture. Likewise the amount of computer time which is saved by the Display Generator depends upon the type and complexity of the picture.

35

An alternative to the Display Generator as described in this paper is to let the computer do the updating of the boundary definitions from one scan line to the next. In this case the computer would send only segment information to a Line Shading device. A segment refers to a portion of a single scan line (1-D information). The information required for each segment could be packed into 36-45 bits. There would be no need for the Merge or Update hardware and no requirement for the large storage memory since the segments would be used once and then discarded. The cost of such an alternative design would be much less than that of the Display Generator.

A second alternative is to build additional devices to further reduce the calculations required by the computer. Further reduction in the price of logic circuits may make it economical to implement the entire process of generating half-tone pictures with special purpose hardware.

In either case the purpose of this paper is to show that a device can be built which will enable computer generated pictures to be displayed in real time.
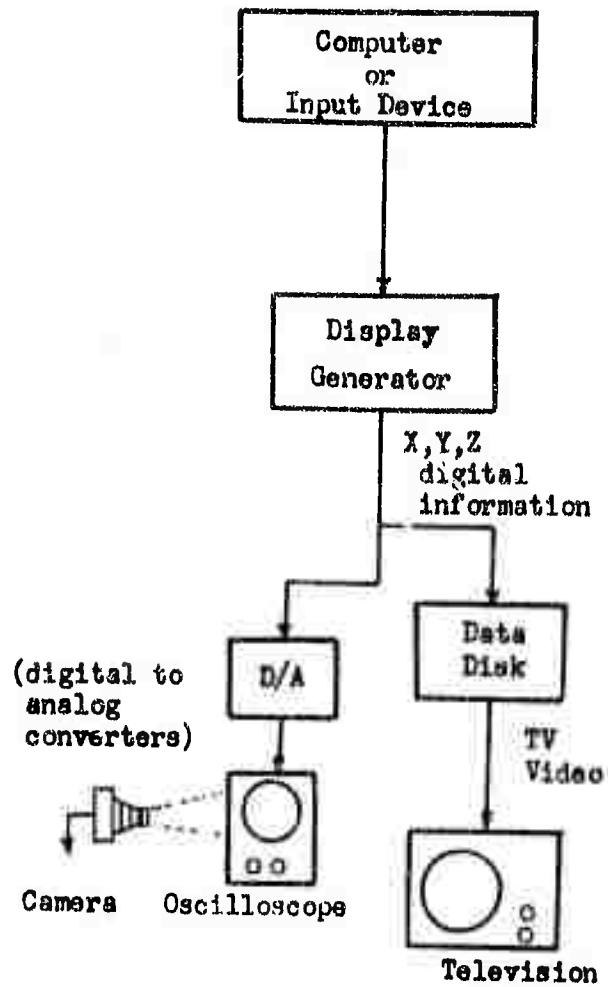
Section 5

LOGIC DRAWINGS

The drawings on the following pages illustrate the organization and operation of the Display Generator. It is intended that these drawings should complement and clarify the description of the Display Generator in section two of this paper.

An explanation of the symbols used in the drawings is given on pages 52 and 53. A listing of all register names and the pages on which they appear may be found on page 54. A listing of the defined signals and the page on which they are defined may be found on pages 55 and 56.

LIST OF LOGIC DRAWINGS

HALF-TONE DISPLAY SYSTEM

Computer or Input Device → Display Generator → X,Y,Z digital information

(digital to analog converters)
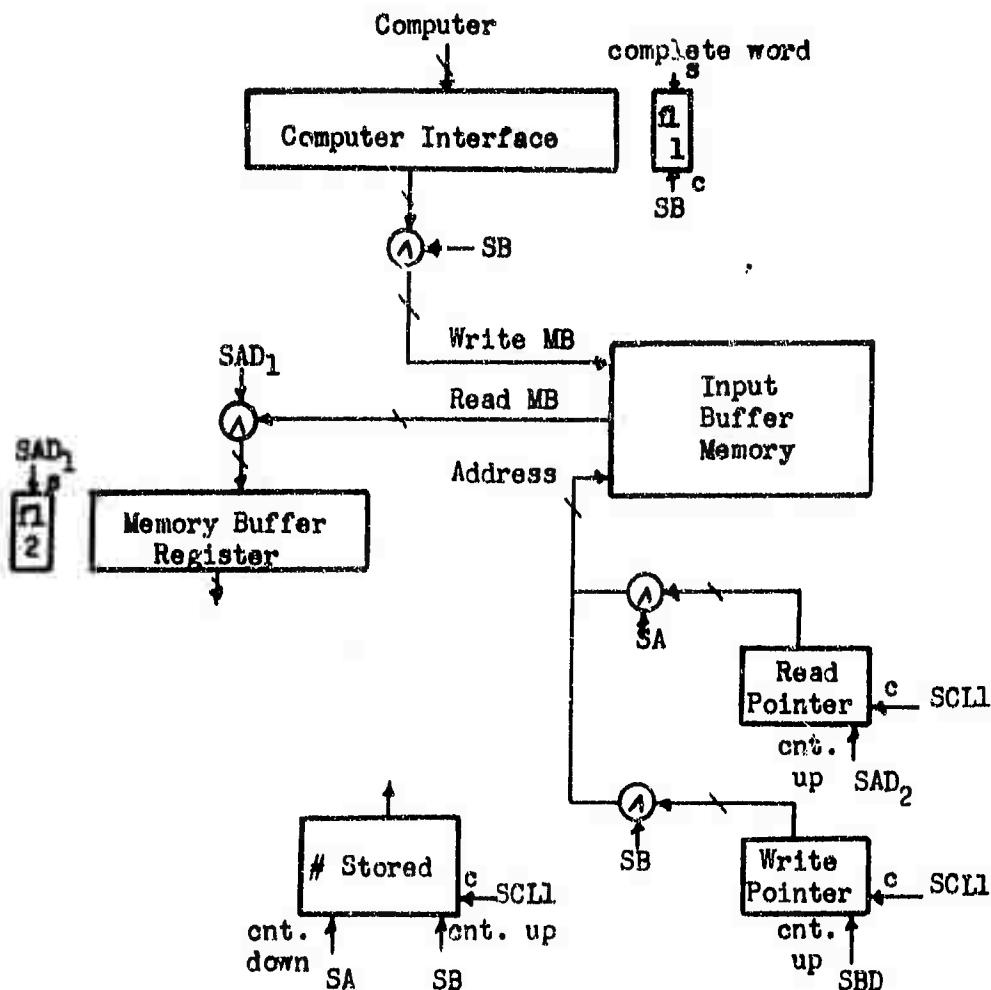
D/A → Oscilloscope / Camera

Data Disk → TV Video → Television

DISPLAY GENERATOR

Input Buffer

$$SA = \text{Cycle A} \wedge \overline{\text{Empty}} \wedge (\overline{\text{fl-2}} \vee SK\;)$$
$$SAD_1 = SA \text{ delayed}$$
$$SAD_2 = SAD_1 \text{ delayed}$$
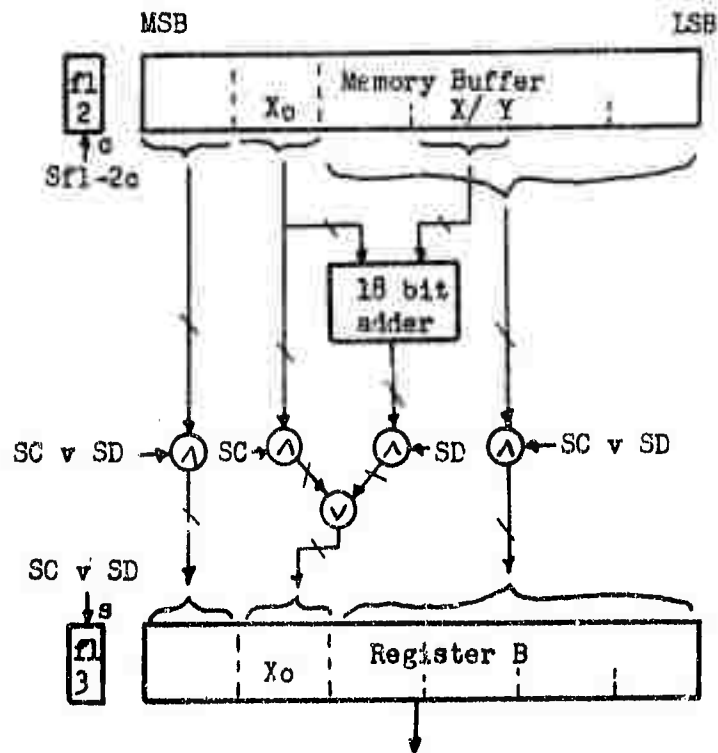$$SB = \text{Cycle A} \wedge \text{fl-1} \wedge \overline{\text{Full}} \wedge (\text{Empty} \vee (\text{fl-2} \wedge \overline{SK^\cdot}))$$
$$SBD = SB \text{ delayed}$$
Empty = # Stored is zero
Full = # Stored equals the size of the Input Buffer Memory
SCL1 = System Clear (manual push button)

$$SC = \text{Cycle } A \wedge f1\text{-}2 \wedge (\overline{f1\text{-}3} \vee SK) \wedge *bit\text{-}0 \wedge (*bit\text{-}5 \vee \overline{**bit\text{-}54})$$

$$SD = \text{Cycle } A \wedge f1\text{-}2 \wedge (\overline{f1\text{-}3} \vee SK) \wedge *bit\text{-}0 \wedge (\overline{*bit\text{-}5} \wedge **bit\text{-}54)$$

$$Sf1\text{-}2c = SCL1 \vee (\overline{SAD}_1 \wedge (SC \vee SD \vee SP \vee SR))$$
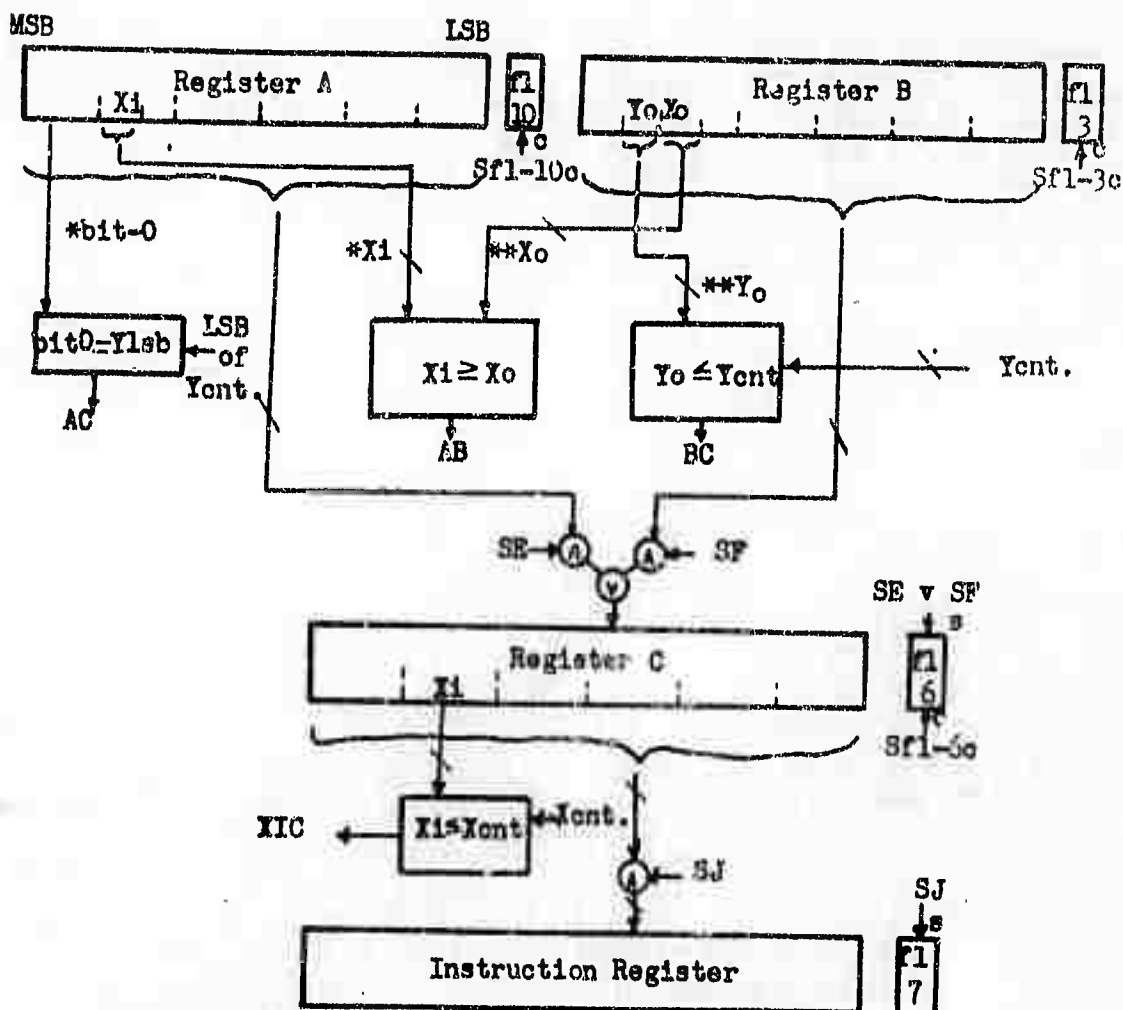
* refers to the Memory Buffer Register

** refers to Register B

Memory Buffer, Registers D1, D2

MSB LSB

Memory Buffer

f1-2

Sf1-2c

SP → ∧

SP

f1-4

Sf1-4c

Register D_1

SQ → ∧   ∧ ← SR

∨

SQ ∨ SR

f1-5

Sf1-5c

Register D_2 Shift 9 bits   SS

Int-3

Shift #

0 to 10 counter

reset   cnt. up
SQ ∨ SR   SS
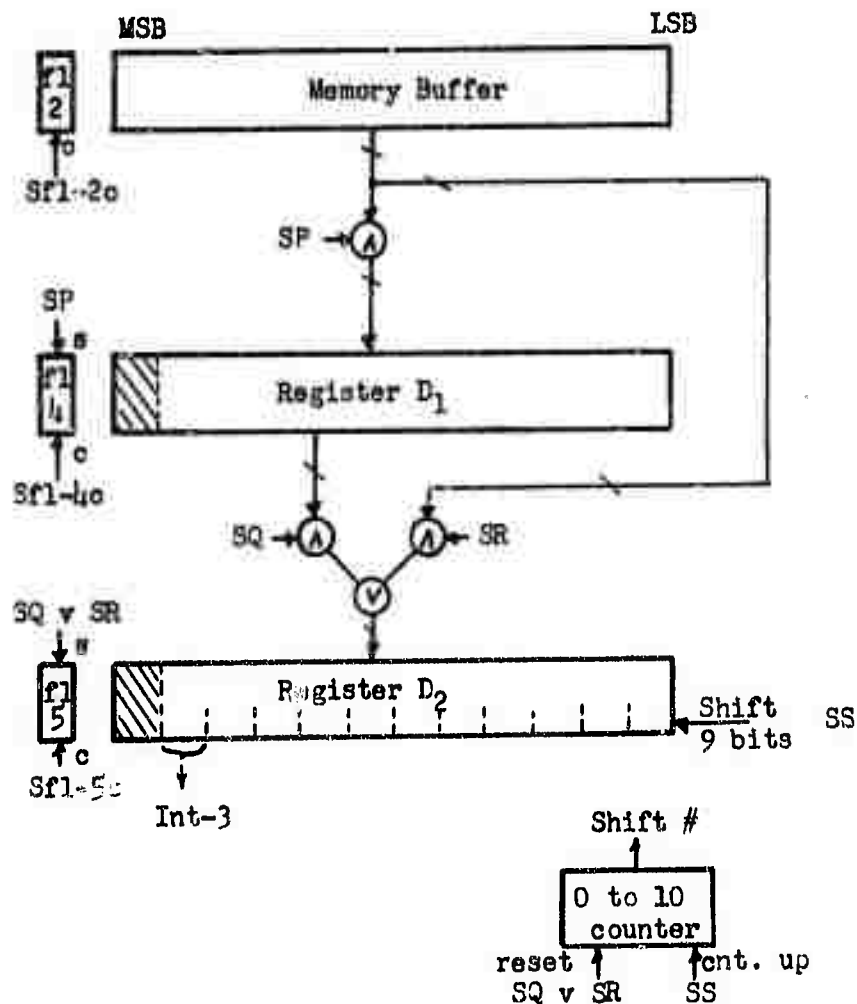
$SP = \text{Cycle A} \wedge \text{f1-2} \wedge *\text{bit-0} \wedge (\text{f1-4} \wedge \text{Lxt}) \vee (\overline{\text{f1-4}} \wedge \text{f1-5} \wedge \overline{\text{Lxt}})$

$SR = \text{Cycle A} \wedge \text{f1-2} \wedge *\text{bit-0} \wedge (\overline{\text{f1-5}} \vee (\text{f1-5} \wedge \text{Lxt}))$

$SQ = \text{Cycle A} \wedge \text{f1-4} \wedge \text{Lxt}$

$SS = \text{Cycle A} \wedge \text{f1-5} \wedge \text{RIP} \wedge \overline{\text{Lxt}}$

$\text{Sf1-4c} = \text{SCL1} \vee (SQ \wedge \overline{SP})$

$\text{Sf1-5c} = \text{RST} \vee (\text{Cycle A} \wedge \text{Lxt} \wedge \overline{SQ} \wedge \overline{SR})$

$\text{Lxt} = \text{Shift } \# = 10$

$\text{RIP} = \text{Request for intensity value from Mode Control}$

$\text{RST} = \text{SCL1} \vee (\text{Shift } \# \neq 0 \wedge \text{SJ})$

$\text{Error1} = \text{RIP} \wedge \overline{\text{f1-5}}$

$*$ Refers to the Memory Buffer

43

$AB = *Xi \geq **X_0$

$AC = *bit-0 = LSB$ of Ycnt.

$BC = **Yo \leq Ycnt.$

$SE = Cycle\ A \wedge f1-10 \wedge AC \wedge (\overline{BC} \vee \overline{AB}) \wedge \overline{STPMRG} \wedge (XIC \vee \overline{f1-6})$

$SF = Cycle\ A \wedge f1-3 \wedge BC \wedge (\overline{AC} \vee AB) \wedge \overline{STPMRG} \wedge (XIC \vee \overline{f1-6})$

$SJ = Cycle\ A \wedge f1-6 \wedge XIC$

$SK = f1-3 \wedge BC \wedge (\overline{AC} \vee AB) \wedge \overline{STPMRG} \wedge (XIC \vee \overline{f1-6})$

$STPMRG = \overline{f1-3} \wedge SW5$

$Sf1-3c = CL1 \vee (SF \wedge \overline{f1-2})$

$Sf1-10c = CL1 \vee (SE \wedge \overline{Memory\ A\ empty})$

$Sf1-6c = CL1 \vee (SJ \wedge \overline{SE} \wedge \overline{SF})$

$XIC = ***Xi \leq Xcnt.$

Xcnt. = X scan position counter (X out)

Ycnt. = Y scan position counter (Y out)

* refers to Register A

** refers to Register B

*** refers to Register C

44

Updating Process
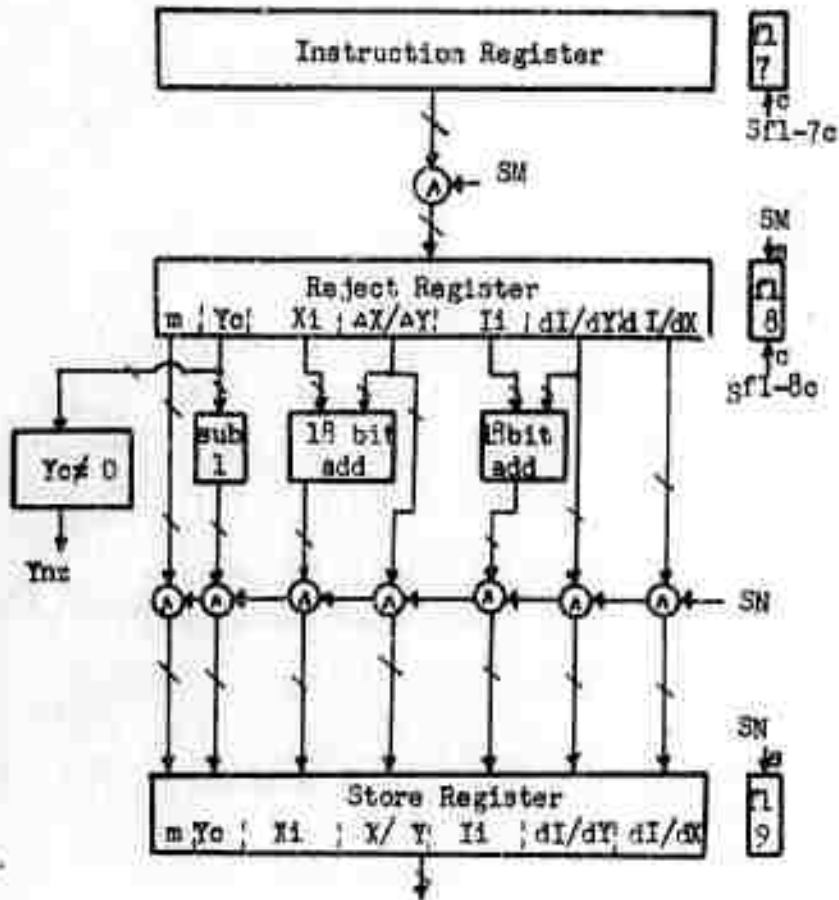
$$SM = \text{Cycle } B \wedge f1\text{-}7$$
$$SN = \text{Cycle } B \wedge f1\text{-}8 \wedge Ynz$$
$$Sf1\text{-}7c = CL1 \vee SM$$
$$Sf1\text{-}8c = CL1 \vee (SN \wedge \overline{f1\text{-}7})$$
$$Ynz = *Yc \neq 0$$

Update Processing
$$**Yc = *Yc-1$$
$$**Xi = *Xi + *\Delta X/\Delta Y$$
$$**Ii = *Ii + *dI/dY$$
$$**m = *m \text{ (except that } **bit\text{-}0 = LSB \text{ of } Ycnt.)$$

* refers to the Reject Register
** refers to the Store Register

45

Register A , Memory A
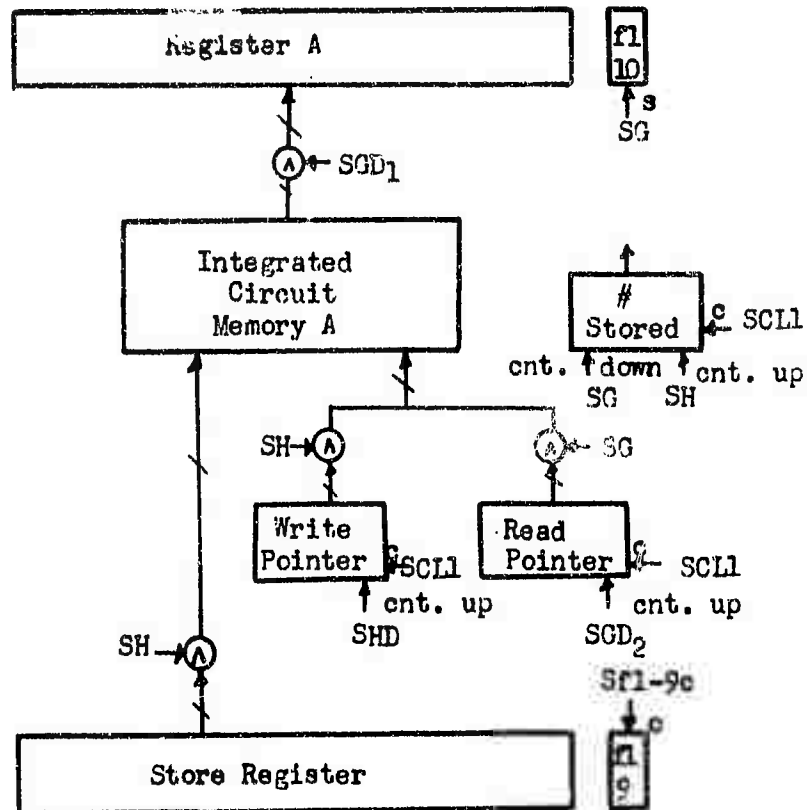
$$SG = \text{Cycle A} \land \overline{\text{EmptyA}} \land (\overline{\text{f1-10}} \lor SE)$$

$SGD_1 = SG$ delayed

$SGD_2 = SGD_1$ delayed

$$SH = \text{Cycle B} \land \text{f1-9} \land \overline{\text{FullA}}$$

$SHD = SH$ delayed

$Sf1-9c = SCL1 \lor (SH \land \overline{SN})$

EmptyA = # Stored equals zero

FullA = # Stored equals the size of Memory A

$Error2 = \text{Cycle B} \land \text{f1-9} \land \text{FullA}$

46

DISPLAY MODES

9 MSB in Boundary Definition
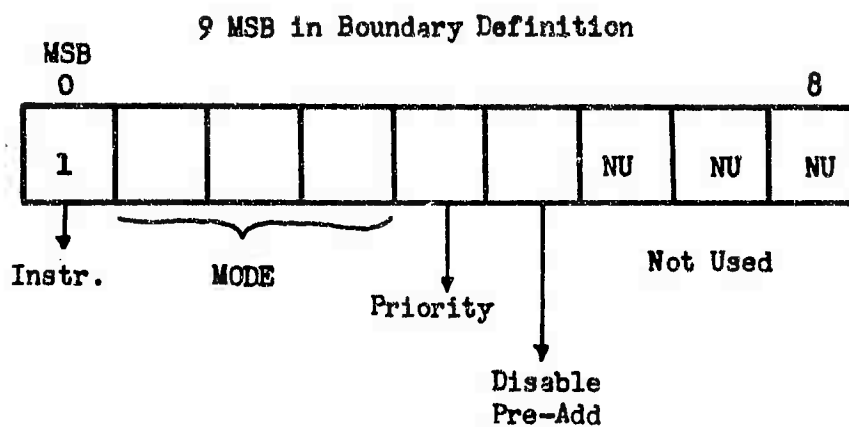
Mode

000 - Turn beam off

001 - Line mode

010 - Linear shading (Initilize shader)

011 - Return to mode 010 (do not reinitilize shader)

100 - Constant Intensity mode

101 - Constant Intensity for Count (then return to mode 010)

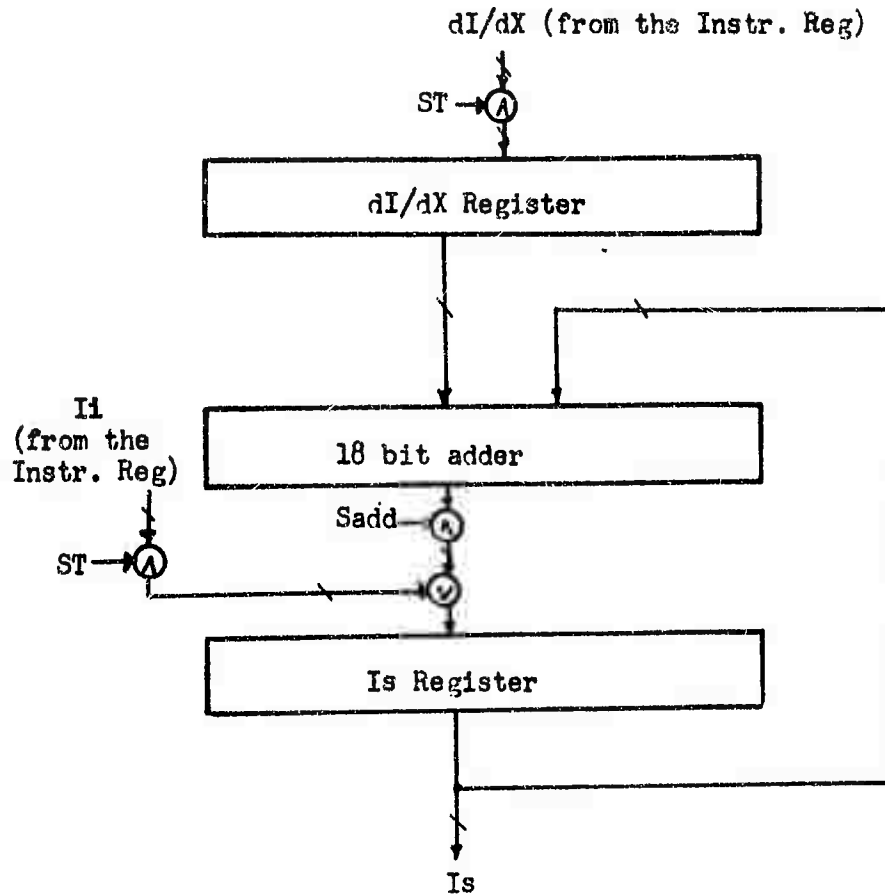110 - 0-D mode (display point information)

111 - End of frame synchronization

Mode Control

Intensity Function Control
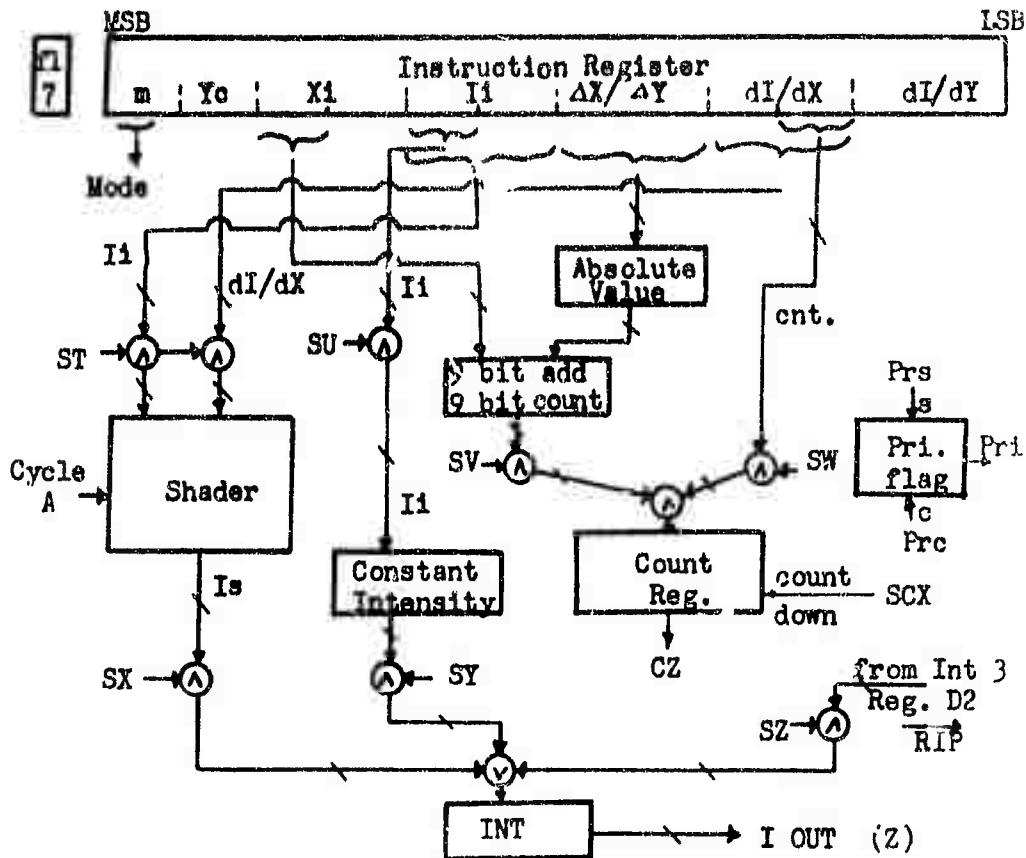
ST = Cycle B ∧ Mode 010 ∧ f1-7
SU = Cycle B ∧ f1-7 ∧ $\overline{Pri}$
SV = Mode 001 ∧ Cycle A ∧ f1-8
SW = Cycle B ∧ Mode 101 ∧ f1-7
SX = Cycle A ∧ ((Mode 01X ∧ $\overline{Pri}$) v (Mode 101 ∧ CZ)) ∧ $\overline{Flgs}$
SY = Cycle A ∧ (Pri v Mode 100 v (Mode 101 ∧ $\overline{CZ}$) v (Mode 001 ∧ $\overline{CZ}$)) ∧ $\overline{Flgs}$
SZ = Cycle A ∧ Mode 110 ∧ $\overline{Pri}$ ∧ Flgs
SCX = Cycle A ∧ $\overline{CZ}$ ∧ Flgs
CZ = equals 1 if Count Reg. equals zero
Prs = f1-7 ∧ Cycle B ∧ Instr. Reg. bit-4
Prc = CL1 v (CZ ∧ Cycle A)
RIP = Cycle A ∧ Mode 110

Shader

dI/dX (from the Instr. Reg)

ST → Ⓐ

```
┌─────────────────────────────────────┐
│           dI/dX Register            │
└─────────────────────────────────────┘
```

Ii
(from the
Instr. Reg)

```
┌─────────────────────────────────────┐
│            18 bit adder             │
└─────────────────────────────────────┘
```

ST → Ⓐ

Sadd — Ⓡ

Ⓥ

```
┌─────────────────────────────────────┐
│             Is Register             │
└─────────────────────────────────────┘
```

Is

Sadd = Cycle A ∧ ($\overline{f1-8}$ v $\overline{Mode\ 010}$) ∧ $\overline{Flgs}$

ST = Cycle B ∧ Mode 010 ∧ f1-7

49

X count, Y count, Clock
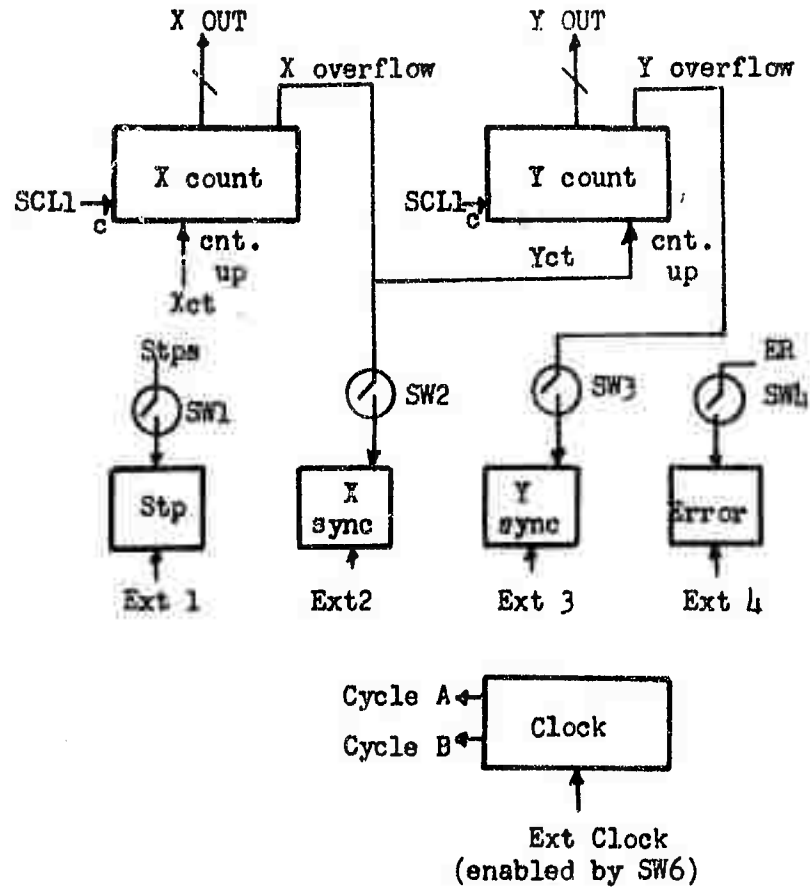
$$Xct = \text{Cycle A} \wedge \overline{\text{Stp}} \wedge \overline{\text{X sync}} \wedge \overline{\text{Y sync}} \wedge \overline{\text{Error}}$$
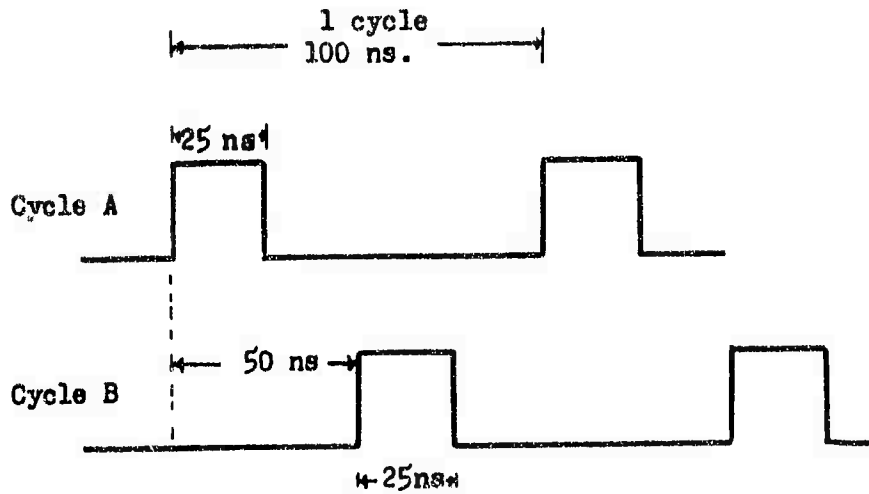
$$ER = \text{Error 1} \vee \text{Error 2} \vee (\text{STPMRG} \wedge \text{SW5})$$

$$Stps = \text{Cycle B} \wedge \text{Mode 111} \wedge \text{f1-7}$$

$$Flgs = \text{Stp} \vee \text{X sync} \vee \text{Y sync} \vee \text{Error}$$
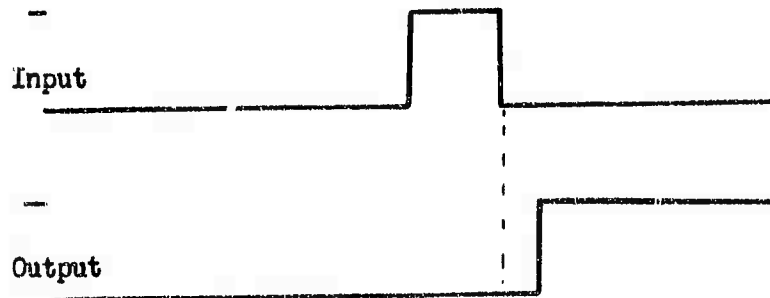
Clock Cycles A and B are either generated internally or are produced by an External clock.

Timing

Clock Signals

Flip Flop trigering

The output of the flip flop (register) will

change after the trailing edge of the input signal

Symbols

∧   Boolean AND symbol

Ⓐ   A digital logic gate which performs an AND function

∨   Boolean OR symbol

Ⓥ   A digital logic gate which performs an OR function

↘   An information or signal path

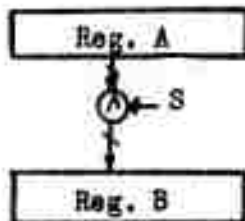↘↘   A number of parallel information or signal paths

A single flip flop referred to as a flag

s means set to 1   c means clear to 0

fl-X refers to the 1 output of flag X
$\overline{\text{fl-X}}$ refers to the 0 output of flag X

These symbols indicate two registers and a jam transfer of the contents of Register A into Register B on the trailing edge of signal S.
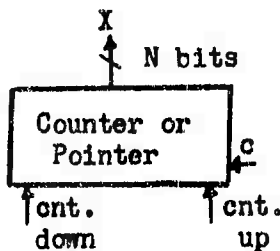
This symbol indicates a parallel jam counter (all outputs change at the same time).

X = the output of the counter

reset
c } → clear or reset the counter to zero

cnt. up → increment the counter by one

cnt. down → decrement the counter by one

52

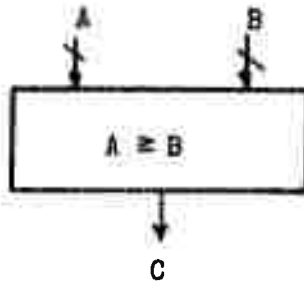Logic functions are described by Boolean equations.
Some logic functions are also shown as boxes.

In this case A and B are the inputs and represent
magnitudes. C is the output and is equal to 1 if $A \gtreqless B$.
C equals 0 if $A \neq B$.

REGISTER LISTING

Register Name	Pages Where Register Appears
Register A	40, 44, 46
Register B	40, 42, 44
Register C	40, 44
Constant Intensity Register	40, 48
Count Register	48
Register D1	40, 43
Register D2	40, 43
Instr Register	40, 44, 45, 48
INT Register	40, 48
Reject Register	40, 45
Shader	40, 48, 49
dI/dX Register	49
Is Register	49
Store A Register	40, 45, 46
X count Register	40, 50
Y count Register	40, 50
Flags	
fl-1	41
fl-2	41, 42, 43
fl-3	42, 44
fl-4	43
fl-5	43
fl-6	44
fl-7	44, 45, 48
fl-8	45
fl-9	45, 46
fl-10	44, 46

DEFINED SIGNALS

Signal	Page Defined	Signal	Page Defined
AB	44	Sadd	49
AC	44	SB	41
BC	44	SBD	41
Clock A	50	DC	42
Clock B	50	DCL1	41
CZ	48	SCX	48
Empty	41	SD	42
EmptyA	46	SE	44
ER	50	SF	44
Error 1	43	Sf1-2c	42
Error 2	46	Sf1-3c	44
Flgs	50	Sf1-4c	43
Full	41	Sf1-5c	43
FullA	46	Sf1-6c	44
LXT	43	Sf1-7c	45
Mode	47	Sf1-8c	45
Prc	48	Sf1-9c	46
Prs	48	Sf1-10c	44
RIP	43, 48	SG	46
RST	43	SGD1	46
SA	41	SGD2	46
SAD1	41	SH	46
SAD2	41	SHD	46

DEFINED SIGNALS CONTINUED

Signal	Page Defined	Signal	Page Defined
SJ	44	SV	48
SK	44	SW	48
SM	45	SX	48
SN	45	SY	48
SP	43	SZ	48
SQ	43	Xcnt	44
SR	43	XCT	50
SS	43	XIC	44
ST	48, 49	X out	44
STPMRG	44	Ycnt	44
Stps	50	Y out	44
SU	48	Ynz	45

BIBLIOGRAPHY

1. Fink, Donald G. Television Engineering. 2nd ed. New York: McGraw Hill, 1952.

2. Romney, Gordon W., et al. "Real-Time Display of Computer-Generated Half-Tone Perspective Pictures," IFIP Congress 68. (August 1968), pp. E72-E77.

3. Warnock, John E. A Hidden Line Algorithm For Halftone Picture Representation. Technical Report 4-5. Salt Lake City, Utah: University of Utah, 1968.

4. Warnock, John E. A Hidden Surface Algorithm For Computer Generated Halftone Pictures. Technical Report 4-15. Salt Lake City, Utah: University of Utah, 1969.

5. Wylie, C., et al. "Half-Tone Perspective Drawings by Computer," AFIPS Conference Proceedings, Fall Joint Computer Conference, XXXI (1967), pp. 49-58.